

Argonne National Laboratory

COMPARISON OF SEVERAL ADAPTIVE
NEWTON-COTES QUADRATURE ROUTINES
IN EVALUATING DEFINITE INTEGRALS
WITH PEAKED INTEGRANDS

by

K. E. Hillstrom

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

MEMBERS OF ARGONNE UNIVERSITIES ASSOCIATION

The University of Arizona	Kansas State University	The Ohio State University
Carnegie-Mellon University	The University of Kansas	Ohio University
Case Western Reserve University	Loyola University	The Pennsylvania State University
The University of Chicago	Marquette University	Purdue University
University of Cincinnati	Michigan State University	Saint Louis University
Illinois Institute of Technology	The University of Michigan	Southern Illinois University
University of Illinois	University of Minnesota	University of Texas
Indiana University	University of Missouri	Washington University
Iowa State University	Northwestern University	Wayne State University
The University of Iowa	University of Notre Dame	The University of Wisconsin

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America

Available from

Clearinghouse for Federal Scientific and Technical Information

National Bureau of Standards, U. S. Department of Commerce

Springfield, Virginia 22151

Price: Printed Copy \$3.00; Microfiche \$0.65

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

COMPARISON OF SEVERAL ADAPTIVE NEWTON-COTES
QUADRATURE ROUTINES IN EVALUATING DEFINITE
INTEGRALS WITH PEAKED INTEGRANDS

by

K. E. Hillstrom

Applied Mathematics Division

November 1968

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	5
I. INTRODUCTION	5
II. QUADRATURE SCHEMES USED	5
III. INVESTIGATIONS CONDUCTED	6
IV. COMPUTATIONS	6
V. RESULTS	7
APPENDIX. FORTRAN Listings of the Comparison Routine, the Integrand and Integral Evaluation Subroutines, and the Five Adaptive Quadrature Subroutines	
1. Routine for Comparing Adaptive Newton-Cotes Quadrature Routines	11
2. Adaptive Quadrature Routine Based on Newton-Cotes 3-point Rule	14
3. Adaptive Quadrature Routine Based on Newton-Cotes 5-point Rule	16
4. Adaptive Quadrature Routine Based on Newton-Cotes 7-point Rule	18
5. Adaptive Quadrature Routine Based on Newton-Cotes 9-point Rule	20
6. Adaptive Quadrature Routine Based on Newton-Cotes 11-point Rule	23
REFERENCES	26

LIST OF FIGURES

<u>No.</u>	<u>Title</u>	<u>Page</u>
1.	Performance Comparison in Approximating $\int_{-1}^1 (x^2 + p^2)^{-1} dx$ for p Ranging from 1 to 10^{-4} and ϵ Ranging from 1 to 10^{-8}	8
2.	Performance Comparison in Approximating $\int_0^1 x/(1.1 - x)^p dx$ for p Ranging from 1 to 9 and ϵ Ranging from 1 to 10^{-8}	8
3.	Performance Comparison in Approximating $\int_0^3 x^p(x-2)(x-3) dx$ for p Ranging from 1 to 18 and ϵ Ranging from 1 to 10^{-8}	8
4.	Error Curves of the Adaptive Newton-Cotes $(2N+1)$ -point Rules in Approximating $I_1 = \int_{-1}^1 (x^2 + 10^{-8})^{-1} dx$	9
5.	Error Curves of the Adaptive Newton-Cotes $(2N+1)$ -point Rules in Approximating $I_2 = \int_0^1 x/(1.1 - x)^9 dx$	9
6.	Error Curves of the Adaptive Newton-Cotes $(2N+1)$ -point Rules in Approximating $I_3 = \int_0^3 x^{18}(x-2)(x-3) dx$	10

COMPARISON OF SEVERAL ADAPTIVE NEWTON-COTES
QUADRATURE ROUTINES IN EVALUATING DEFINITE
INTEGRALS WITH PEAKED INTEGRANDS

by

K. E. Hillstrom

ABSTRACT

This report compares the performance of five different adaptive quadrature schemes, based on Newton-Cotes $(2N+1)$ -point rules ($N = 1, 2, 3, 4, 5$), in approximating the sets of definite integrals

$$\int_{-1}^1 (x^2 + p^2)^{-1} dx, \int_0^1 x/(1.1 - x)^p dx, \text{ and } \int_0^3 x^p(x - 2)(x - 3) dx$$

with relative accuracy ϵ .

I. INTRODUCTION

High-degree Newton-Cotes quadrature rules have seldom been used in practice because they occasionally fail to converge or they contain weights of different signs. This report shows, by numerical experiment only, that there are certain quadratures for which moderate degree Newton-Cotes rules, used adaptively, may be superior to other rules based on equally spaced abscissas.

McKeeman¹⁻³ and Davis and Rabinowitz⁴ describe the use of quadrature rules in an adaptive manner by means of algorithms. Lyness⁵ gives a thorough description of the Adaptive Simpson Rule, together with suggested modifications.

II. QUADRATURE SCHEMES USED

The schemes used in these investigations are adaptive Newton-Cotes rules of degree 3, 5, 7, 9, and 11, incorporating the modifications of types 1, 2, and 3 described in Ref. 5, adjusted for use with the particular rule. The results are, therefore, of polynomial degree 5, 7, 9, 11, and 13, respectively.

In particular, if ϵ is the total absolute error allowed and

$$Q_N[a, a+h] f(x) = \sum_{j=1}^{2N+1} a_j f(x_j)$$

is the $(2N+1)$ Newton-Cotes Rule, convergence is achieved over the interval $[a, a+h]$ if for

$$\Delta = Q_N[a, a+h] f(x) - \left(Q_N\left[a, a + \frac{h}{2}\right] f(x) + Q_N\left[a + \frac{h}{2}, a+h\right] f(x) \right),$$

$$|\Delta| \leq \frac{\left(2^{2N+1} - 1\right) \epsilon}{2^r},$$

where $h = (B - A) 2^{-r}$. If this convergence criterion is not satisfied and if $r < 30$, the interval is bisected, r is replaced by $r + 1$, and, after function evaluations at the new mesh points, the above test is repeated. If the convergence criterion is satisfied or if $r = 30$,

$$Q_N\left[a, a + \frac{h}{2}\right] f(x) + Q_N\left[a + \frac{h}{2}, a+h\right] f(x) + \Delta / \left(2^{2N+1} - 1\right)$$

is accepted as an approximation for the integral over $[a, a+h]$, this approximation being of degree $2N + 3$. Finally, these component approximations and error estimates are summed to obtain a final or total approximation over $[A, B]$.

III. INVESTIGATIONS CONDUCTED

Numerical integrations were carried out using the five different adaptive Newton-Cotes rules in approximating three sets of definite integrals. In each case, an input parameter ϵ prescribes the error. A routine is termed the most efficient if its result satisfies the error criterion ϵ while requiring the fewest function evaluations. The actual error in each result is usually much smaller than ϵ .

IV. COMPUTATIONS

All the computations conducted were performed on an IBM 360/50-75 in double-precision floating point.

The sets of definite integrals

$$I_1 = \int_{-1}^1 (x^2 + p^2)^{-1},$$

$$I_2 = \int_0^1 x/(1.1 - x)^p dx,$$

and

$$I_3 = \int_0^3 x^p(x - 2)(x - 3) dx$$

have been evaluated with p ranges $(1, 10^{-4})$, $(1, 9)$, and $(1, 16)$, respectively, and ϵ ranging from 1 to 10^{-8} .

For small p , the I_1 integrand has a peak of height p^{-2} at the origin and is approximately 1 at the end points, and I_1 is approximately equal to πp^{-1} .

For large p , the I_2 integrand has a maximum of 10^p at the upper limit and is zero at the origin, and I_2 is approximately equal to $10^{p-1}/(p - 1)$.

For large p , the I_3 integrand has a maximum near 2, has a minimum near 3, and is zero at the origin and at $x = 2$ and $x = 3$, and I_3 is approximately equal to $-3^{p+2}/p^2$.

The appendix presents FORTRAN listings of the comparison routine, the integrand and integral evaluation subroutines, and the five adaptive quadrature subroutines.

V. RESULTS

Results using the adaptive Newton-Cotes rules for the quadratures I_1 , I_2 , and I_3 are displayed in Figs. 1, 2, and 3, respectively, and are to the required accuracy ϵI . If the point (p, ϵ) lies in the zone numbered N , the adaptive Newton-Cotes $(2N+1)$ -point rule is the most efficient of those tested.

The actual demarcation lines between zones are not regular in Figs. 1 and 2. These irregularities are partly due to the fact that the number of points used by the adaptive Newton-Cotes rule of degree $2N + 1$ in approximating an integral over $(a, a+h)$ is restricted to numbers of the form $8kN + 1$ ($k = 1, 2, 3, \dots$). If the integrand function is altered slightly, the demarcation line is different in detail, but has the same general configuration. In Figs. 1 and 2, the "buffer zones" between distinct zones indicate the general width of these irregularities.

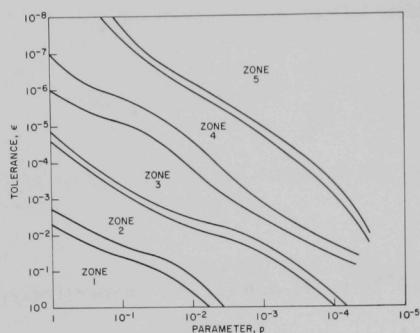


Fig. 1. Performance Comparison in Approximating $\int_{-1}^1 (x^2 + p^2)^{-1} dx$ for p Ranging from 1 to 10^{-4} and ϵ Ranging from 1 to 10^{-8} . If the point (p, ϵ) lies in the zone numbered N , the adaptive Newton-Cotes $(2N+1)$ -point rule is the most efficient of those tested.

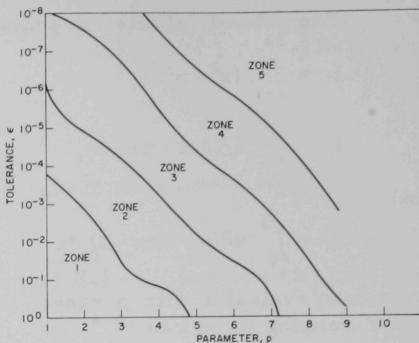


Fig. 2. Performance Comparison in Approximating $\int_1^9 x/(1.1-x)^p dx$ for p Ranging from 1 to 9 and ϵ Ranging from 1 to 10^{-8} . If the point (p, ϵ) lies in the zone numbered N , the adaptive Newton-Cotes $(2N+1)$ -point rule is the most efficient of those tested.

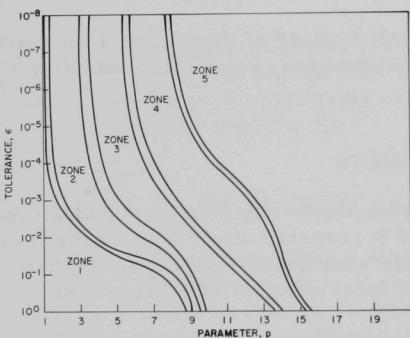


Fig. 3
Performance Comparison in Approximating $\int_0^3 x^p(x-2)(x-3) dx$ for p Ranging from 1 to 16 and ϵ Ranging from 1 to 10^{-8} . If the point (p, ϵ) lies in the zone numbered N , the adaptive Newton-Cotes $(2N+1)$ -point rule is the most efficient of those tested.

These results are of interest because they indicate that if the integrand has a high, sharp peak, or if great accuracy is required, an adaptive high-degree rule is most efficient. This is in contradistinction to the more familiar state of affairs in which sharp peaks are associated with inefficient polynomial approximations and the use of low-degree rules.

In addition, error curves of the adaptive Newton-Cotes rules for the quadratures I_1 , I_2 , and I_3 and a particular value of p are displayed in Figs. 4, 5, and 6, respectively. The error curves of the adaptive Newton-Cotes $(2N+1)$ -point rules are labeled EN; the projections of the pairs of intersection points (p_{N-1}, p_N) onto the ϵ axis define, for fixed p , the

$\log \epsilon$ interval over which the $(2N+1)$ -point rule is most efficient, in terms of points M required; and the lines S₁, S₂, S₃, S₄, and S₅ with slopes 6, 8, 10, 12, and 14, respectively, indicate the rate of convergence of these rules.

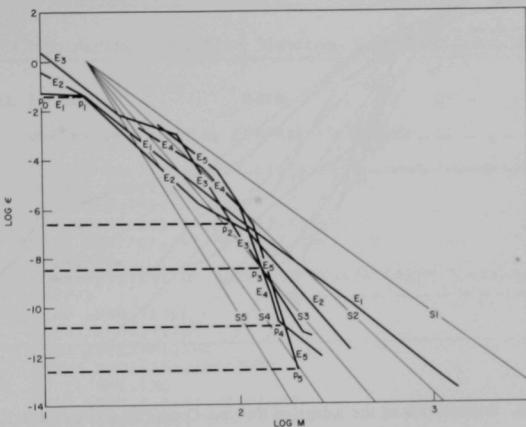


Fig. 4. Error Curves of the Adaptive Newton-Cotes $(2N+1)$ -point Rules in

Approximating $I_1 = \int_{-1}^1 (x^2 + 10^{-8})^{-1} dx$. A point (M, ϵ) located

on line E_N between intersection points (P_{N-1}, P_N) indicates that the $2N+1$ adaptive Newton-Cotes routine with input error ϵE_1 required the least number M of function evaluations of the routines tested.

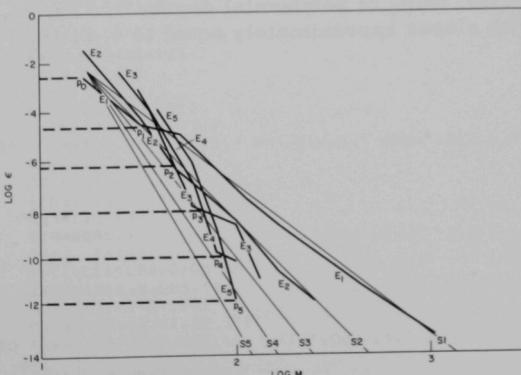


Fig. 5. Error Curves of the Adaptive Newton-Cotes $(2N+1)$ -point Rules in

Approximating $I_2 = \int_0^1 x/(1.1-x)^9 dx$. A point (M, ϵ) located

on line E_N between intersection points (P_{N-1}, P_N) indicates that the $2N+1$ adaptive Newton-Cotes routine with input error ϵE_1 required the least number M of function evaluations of the routines tested.

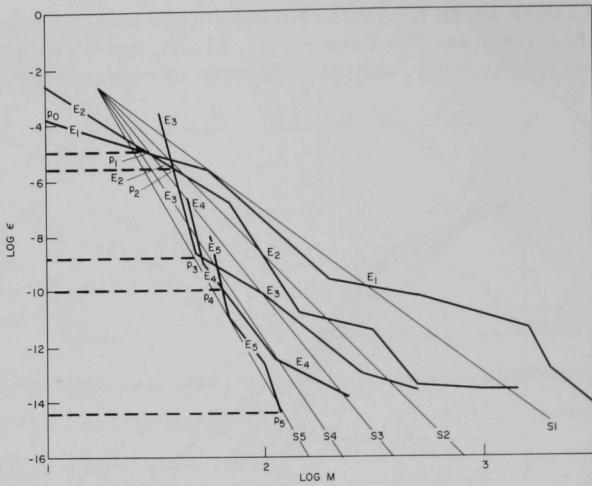


Fig. 6. Error Curves of the Adaptive Newton-Cotes $(2N + 1)$ -point Rules in Approximating $I_3 = \int_0^3 x^{18}(x - 2)(x - 3) dx$. A point (M, ϵ) located on line E_N between intersection point (p_{N-1}, p_N) indicates that the $2N + 1$ adaptive Newton-Cotes routine with input error ϵI_3 required the least number M of function evaluations of the routines tested.

These results define (for a particular p) the ϵ interval of greatest efficiency of the $(2N + 1)$ -point rules and demonstrate that the modified Newton-Cotes rules, being of polynomial degree 5, 7, 9, 11, and 13 have error curves with slopes approximately equal to 6, 8, 10, 12, and 14, respectively.

APPENDIX

FORTRAN Listings of the Comparison Routine, the Integrand
and Integral Evaluation Subroutines, and the Five
Adaptive Quadrature Subroutines

1. Routine for Comparing Adaptive Newton-Cotes Quadrature Routines

FORTRAN IV G LEVEL 1, MOD 1	MAIN	DATE = 68288 16/56/53
-----------------------------	------	----------------------------

```

C     QUADRATURE ROUTINE COMPARISON DRIVER
C
0001    DOUBLE PRECISION A,B,EI,FE,PI,DP,FP,P,EPS,QI,RI,RE,RES,QUAD,T,EVAL
0002    DOUBLE PRECISION DE
0003    COMMON P,IFI,IFC,ICC
0004    DIMENSION IVC(5),QDR(6),TITLE(18)
0005    DATA QDR//2',4',6',8',0','X'
0006    PRINT 15
0007    15 FORMAT('1TEST OF ADAPTIVE NEWTON COTES 2,4,6,8,10 HAVIE AND KUMBER
X'//')
0008    1 READ 1000,TITLE
0009    PRINT 1000,TITLE
0010    READ 1001,IFI,INC
0011    PRINT 1001,IFI,INC
0012    DO 11 N=1,INC
0013    READ 1002,A,B,EI,DE,PI,DP
0014    PRINT 1006,A,B+EI,DE,PI,DP
0015    READ 1004,FE,FP,IE,IP,ICVC
0016    PRINT 1007,FE,FP,IE,IP,ICVC
0017    DO 10 M=1,ICVC
0018    READ (5,1003) (IVC(I),I=1,5)
0019    READ 1005,K,ICC
0020    IFCALL=1
0021    XMAX=FP
0022    XPINCH=(FP-PI)/10.000
0023    YMAX=FE
0024    YMINT=EI
0025    XMINT=PI
0026    XP=PI
0027    GO TO (2,3),K
0028    2 P=PI
0029    GO TO 4
0030    3 P=10.000**(-PI)
0031    4 DO 9 L=1,IP
0032    YP=EI
0033    EPS=10.000**(-EI)
0034    PRINT 20,A,B,EPS,P
0035    20 FORMAT(A=',D12.5,' B=',D12.5,' EPS=',D12.5,' P=',D12.5,//)
0036    RI=VAL(A,B)
0037    DO 8 J=1,IE
0038    IFCs=2147483647
0039    RES=1.00+74
0040    SYM=UDR(6)
0041    DO 7 I=1,5
0042    IF(IVC(I).EQ.0)GO TO 7
0043    QI=QUAD(A,B+EPS,I)
0044    RE=DABS(RI-QI)/RI
0045    PRINT 30,QI,RI,RE,IFC
0046    30 FORMAT(' QI=',D22.15,' RI=',D22.15,' RERR=',D22.15,' FCOUNT=',I8,/
X/')
0047    IF(RE.GT.EPS)GO TO 7
0048    IF(IFCS.LT.IFC)GO TO 7
0049    IF(IFCS.GT.IFC)GO TO 7
0050    IF(RES.LE.RE)GO TO 7
0051    6 IFCs=IFC

```

FORTRAN IV G LEVEL 1, MOD 1

MAIN

DATE = 68288

16/56/53

```

0052      RES=RE
0053      SYM=QDR(I)
0054      7 CONTINUE
0055      PRINT 50
0056      50 FORMAT(' **** * **** * **** * **** * **** * **** * **** * **** * //')
0057      GO TO (16,17),K
0058      16 CALL VOLYPL(XP,YP,XMAX,XPINCH,YMAX,0,1,K,BHPARAMETR,BH-LOG(EP),IFC
          XALL,1,1,XMIN,YMIN,SYM,TITLE,1)
          GO TO 18
0059      17 CALL VOLYPL(XP,YP,XMAX,XPINCH,YMAX,0,1,K,BH-LOG(P) ,BH-LOG(EP),IFC
          XALL,1,1,XMIN,YMIN,SYM,TITLE,1)
0060      18 IF(IFCALL.EQ.2)GO TO 12
0061      IFCALL=2
0062      12 EPS=EPS*10.0D0**(-DE)
0063      8 YP=YP+DE
0064      XP=XP+DP
0065      GO TO (19,21),K
0066      19 P=P+DP
0067      GO TO 9
0068      21 P=P*10.0D0**(-DP)
0069      9 CUNTINUE
0070      10 CUNTINUE
0071      11 CUNTINUE
0072      CALL PLOT(15.,0,-3)
0073      CALL VOLYPL(XP,YP,XMAX,XPINCH,YMAX,-1,1,K,BHPARAMETR,BH-LOG(EP),IFC
          XCALL,1,1,XMIN,YMIN,SYM,TITLE,1)
0074      STOP
0075      1000 FORMAT(18A4)
0076      1001 FORMAT(2I2)
0077      1002 FORMAT(6D12.5)
0078      1003 FORMAT(5I1)
0079      1004 FORMAT(2D12.5,I3,I3,I2)
0080      1005 FORMAT(I3,I2)
0081      1006 FORMAT(' A=',D12.5,' B=',D12.5,' IEP=',D12.5,' DEP=',D12.5,' IP=',
          XD12.5,' DP=',D12.5)
0082      1007 FORMAT(' FE=',D12.5,' FP=',D12.5,' EC= ',I3,' PC= ',I3,' VC= ',I2,
          X//)
0083      END
0084

```

FORTRAN IV G LEVEL 1, MOD 1

MAIN

DATE = 68270

12/16/41

```

C      QUADRATURE ROUTINE SELECTOR
C
0001      FUNCTION QUAD(A,B,EPS,I)
0002      DOUBLE PRECISION A,B,EPS,EP,FCN,QUAD,ANC2,ANC4,ANC6,ANC8,ANC10,P
0003      COMMON P,IFI,IFC,ICC
0004      EXTERNAL FCN
0005      IFC=0
0006      EP=EPS
0007      M=15
0008      N=ICC
0009      GO TO(10,20,30,40,50),I
0010      10 QUAD=ANC2(A,B,EP,M,N,FCN)
0011      RETURN
0012      20 QUAD=ANC4(A,B,EP,M,N,FCN)
0013      RETURN
0014      30 QUAD=ANC6(A,B,EP,M,N,FCN)
0015      RETURN
0016      40 QUAD=ANC8(A,B,EP,M,N,FCN)
0017      RETURN
0018      50 QUAD=ANC10(A,B,EP,M,N,FCN)
0019      RETURN
0020      END

```

FORTRAN IV G LEVEL 1, MOD 1

MAIN

DATE = 68270

12/16/41

```

C      EVALUATES INTEGRAND FUNCTION
C
0001    FUNCTION FCN(X)
0002      DOUBLE PRECISION X,P,FCN
0003      COMMON P,IFI,IFC,ICC
0004      IFC=IFC+1
0005      GO TO(10,20,30),IFI
0006      10 FCN=1.0D0/(X*X+P*P)
0007      RETURN
0008      20 FCN=X/(1.1D0-X)**P
0009      RETURN
0010      30 FCN=X**P*(X-2.0D0)*(X-3.0D0)
0011      RETURN
0012      END

```

FORTRAN IV G LEVEL 1, MOD 1

MAIN

DATE = 68270

12/16/41

```

C      EVALUATES DEFINITE INTEGRAL
C
0001    FUNCTION EVAL(A,B)
0002      DOUBLE PRECISION A,B,P,EVAL,T1,T2,T3,T4,Q1,Q2
0003      COMMON P,IFI,IFC,ICC
0004      GO TO(10,20,30),IFI
0005      10 EVAL=(DATAN(B/P)-DATAN(A/P))/P
0006      RETURN
0007      20 IF(P.EQ.1.0D0) GO TO 24
0008      IF(P.EQ.2.0D0) GO TO 28
0009      Q1=P-1.0D0
0010      Q2=P-2.0D0
0011      T1=(1.1D0-B)**Q2
0012      T2=(1.1D0-B)**Q1
0013      T3=(1.1D0-A)**Q2
0014      T4=(1.1D0-A)**Q1
0015      EVAL=-1.0D0/(Q2*T1)+1.1D0/(Q1*T2)+1.0D0/(Q2*T3)-1.1D0/(Q1*T4)
0016      RETURN
0017      24 T1=DABS(1.1D0-B)
0018      T2=DABS(1.1D0-A)
0019      EVAL=A-B-1.1D0*(DLOG(T1)-DLOG(T2))
0020      RETURN
0021      28 T1=DABS(1.1D0-B)
0022      T2=DABS(1.1D0-A)
0023      EVAL=DLOG(T1)-DLOG(T2)+1.1D0/(1.1D0-B)-1.1D0/(1.1D0-A)
0024      RETURN
0025      30 T1=B*B
0026      T2=A*A
0027      T3=P+1.0D0
0028      EVAL=B**T3*(T1/(P+3.0D0)-5.0D0*B/(P+2.0D0)+6.0D0/(P+1.0D0))-A**T3*
X(T2/(P+3.0D0)-5.0D0*A/(P+2.0D0)+6.0D0/(P+1.0D0))
0029      RETURN
0030      END

```

2. Adaptive Quadrature Routine Based on Newton-Cotes 3-point Rule

```

FORTRAN IV G LEVEL 1, MOD 1           MAIN          DATE = 68270      12/16/41
C      ANC2 INTEGRATION
C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 3 POINT RULE
C
0001   FUNCTION ANC2(A1,B1,EP,N,FUN)
0002   DOUBLE PRECISION ANC2,A1,B1,EP,FUN,A,B,EPS,ABSR,EST,FA,FM,FB,XB,
0003   F1,F2,FBP,EST2,DIFF,EST1,SUM,DAFT,ESUM,TSUM,DA,SX,SA
0004   DOUBLE PRECISION AEST2,FTST,FMAX,AEST1,DELTA,AEST
0005   DIMENSION F2(30),FBP(30),EST2(30),NRTR(30)
0006   DIMENSION AEST2(30),FTST(3),XB(30)
C      THE PARAMETER SETUP FOR THE INITIAL CALL
0007   IF(N.LE.0)GO TO 210
0008   IF(N.GT.3)GO TO 211
0009   A=A1
0010   B=B1
0011   EPS=EP*15.000
0012   ESUM=0.000
0013   TSUM=0.000
0014   LVL=1
0015   DA=B-A
0016   FA=FUN(A)
0017   FM=FUN((A+B)*0.500)
0018   FB=FUN(B)
0019   M=3
0020   FMAX=DABS(FA)
0021   FTST(1)=FMAX
0022   FTST(2)=DABS(FM)
0023   FTST(3)=DABS(FB)
0024   DO 100 I=2,3
0025   IF(FMAX.GE.FTST(I))GO TO 100
0026   FMAX=FTST(I)
0027   100 CONTINUE
0028   EST=(FA+4.000*FM+FB)*DA/6.000
0029   ABSR=(FTST(1)+4.000*FTST(2)+FTST(3))*DA/6.000
0030   AEST=ABSR
0031   1=RECUR
0032   1 SX=(DA/(2.000**LVL))/6.000
0033   F1=FUN((3.000*A+B)/4.000)
0034   F2(LVL)=FUN((A+3.000*B)/4.000)
0035   EST1=SX*(FA+4.000*F1+FM)
0036   FBP(LVL)=FB
0037   XB(LVL)=B
0038   EST2(LVL)=SX*(FM+4.000*F2(LVL)+FB)
0039   SUM=EST1+EST2(LVL)
0040   FTST(1)=DABS(F1)
0041   FTST(2)=DABS(F2(LVL))
0042   FTST(3)=DABS(FM)
0043   AEST1=SX*(DABS(FA)+4.000*FTST(1)+FTST(3))
0044   AEST2(LVL)=SX*(FTST(3)+4.000*FTST(2)+DABS(FB))
0045   ABSR=ABSR-AEST+AEST1+AEST2(LVL)
0046   M=M+2
0047   GO TO (201,200,202),N
0048   200 DELTA=ABSR
0049   GO TO 205
0050   210 PRINT 39
0051   39 FORMAT(' ERROR RETURN-N.LE.0')

```

FORTRAN IV G LEVEL 1, MCD 1

ANC2

DATE = 68270

12/16/41

```

0050      RETURN
0051 211 PRINT 40
0052      40 FORMAT(' ERROR RETURN-N.GT.3')
0053      RETURN
0054 201 DELTA=1.000
0055      GO TO 205
0056 202 DO 203 I=1,2
0057      IF(FMAX.GE.FTST(I))GO TO 203
0058      FMAX=FTST(I)
0059 203 CONTINUE
0060      DELTA=FMAX
0061 205 DAFT=EST-SUM
0062      DIFF=DABS(DAFT)
0063      DAFT=DAFT/15.000
0064      IF(DIFF-EPS*DELTA)6,6,3
0065      3 IF(LVL-30)4,2,2
0066      6 IF(LVL-1)2,4,2
0067      2=UP
0068      2 A=B
0069      ESUM=ESUM+DAFT
0070      TSUM=TSUM+SUM
0071      9 LVL=LVL-1
0072      L=NRTR(LVL)
0073      GO TO (11,12),L
0074      C 11=R1,12=R2
0075      NRTR(LVL)=1
0076      EST=EST1
0077      AEST=AEST1
0078      FB=FM
0079      FM=F1
0080      B=(A+B)/2.000
0081      EPS=EPS/2.000
0082      7 LVL=LVL+1
0083      GO TO 1
0084      11 NRTR(LVL)=2
0085      FA=FB
0086      FM=F2(LVL)
0087      FB=FBP(LVL)
0088      B=XB(LVL)
0089      EST=EST2(LVL)
0090      AEST=AEST2(LVL)
0091      GO TO 7
0092      12 EPS=2.000*EPS
0093      IF(LVL-1) 5,5,9
0094      5 ANC2=TSUM-ESUM
0095      RETURN
0096      END

```

3. Adaptive Quadrature Routine Based on Newton-Cotes 5-point Rule

```

FORTRAN IV G LEVEL 1, MOD 1           MAIN          DATE = 68270      12/16/41
C      ANC4 INTEGRATION
C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 5 POINT RULE
C
0001    FUNCTION ANC4(A1,B1,EP,M,N,FUN)
0002    DOUBLE PRECISION ANC4,A1,B1,EP,FUN,A,B,EPS,ESUM,TSUM,DA,XB,SX,FA
0003    1,F1,FS,F3,F4,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST
0004    21,AEST2,ABSAR,DELTA,DIFF,DAFT,SUM,SA
0005    DIMENSION F2(30),F4(30),FTP(30),FBP(30),FTST(5),EST2(30),NRTR(30)
0006    DIMENSION AEST2(30),XB(30)
0007    C      THE PARAMETER SETUP FOR THE INITIAL CALL
0008    IF(N.LE.0)GO TO 210
0009    IF(N.GT.3)GO TO 211
0010    A=A1
0011    B=B1
0012    EPS=EP*63.000
0013    ESUM=0.000
0014    LVL=1
0015    DA=B-A
0016    FA=FUN(A)
0017    FS=FUN((3.000*A+B)/4.000)
0018    FM=FUN((A+B)*0.500)
0019    FT=FUN((A+3.000*B)/4.000)
0020    FB=FUN(B)
0021    M=5
0022    FMAX=DABS(FA)
0023    FTST(1)=FMAX
0024    FTST(2)=DABS(FS)
0025    FTST(3)=DABS(FM)
0026    FTST(4)=DABS(FT)
0027    FTST(5)=DABS(FB)
0028    DO 100 I=2,5
0029    IF(FMAX.GE.FTST(I))GO TO 100
0030    FMAX=FTST(I)
0031    100 CONTINUE
0032    EST=(7.000*(FA+FB)+32.000*(FS+FT)+12.000*FM)*DA/90.000
0033    ABSAR=(7.000*(FTST(1)+FTST(5))+32.000*(FTST(2)+FTST(4))+12.000*FTS
0034    1T(3))*DA/90.000
0035    AEST=ABSAR
0036    C      1=RECUR
0037    1  SX=(DA/(2.000*LVL))/90.000
0038    F1=FUN((7.000*A+B)/8.000)
0039    F3=FUN((5.000*A+3.000*B)/8.000)
0040    F2(LVL)=FUN((3.000*A+5.000*B)/8.000)
0041    F4(LVL)=FUN((A+7.000*B)/8.000)
0042    EST1=SX*(7.000*(FA+FM)+32.000*(F1+F3)+12.000*FS)
0043    FBPL(LVL)=FB
0044    FTP(LVL)=FT
0045    XB(LVL)=B
0046    EST2(LVL)=SX*(7.000*(FM+FB)+32.000*(F2(LVL)+F4(LVL))+12.000*FT)
0047    SUM=EST1+EST2(LVL)
0048    FTST(1)=DABS(F1)
0049    FTST(2)=DABS(F2(LVL))
0050    FTST(3)=DABS(F3)
0051    FTST(4)=DABS(F4(LVL))

```

FORTRAN IV G LEVEL 1, MCD 1

ANC4

DATE = 68270

12/16/41

```

C048      FTST(5)=DABS(FM)
C049      AEST1=SX*(7.000*(DABS(FA)+FTST(5))+32.000*(FTST(1)+FTST(3))+12.000
C050      X*DABS(FS))
C051      AEST2(LVL)=SX*(7.000*(FTST(5)+DABS(FB))+32.000*(FTST(2)+FTST(4))+1
C052      X2.000*DABS(FT))
C053      ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
C054      M=M+4
C055      GO TO (201,200,202),N
C056      200 DELTA=ABSAR
C057      GO TO 205
C058      210 PRINT 39
C059      39 FORMAT(' ERROR RETURN-N.LE.0')
C060      RETURN
C061      211 PRINT 40
C062      40 FORMAT(' ERROR RETURN-N.GT.3')
C063      RETURN
C064      201 DELTA=1.000
C065      GO TO 205
C066      202 DO 203 I=1,4
C067      IF(FMAX.GE.FTST(I))GO TO 203
C068      FMAX=FTST(I)
C069      203 CONTINUE
C070      DELTA=FMAX
C071      205 DAFT=EST-SUM
C072      DIFF=DABS(DAFT)
C073      DAFT=DAFT/63.000
C074      IF(DIFF-EPS*DELTA)6,6,3
C075      3 IF(LVL-30)4,2,2
C076      6 IF(LVL-1)2+,2
C077      2=UP
C078      C 2 A=B
C079      ESUM=ESUM+DAFT
C080      TSUM=TSUM+SUM
C081      9 LVL=LVL-1
C082      L=NRT(R,LVL)
C083      GO TO (11,12),L
C084      C 11=R1,12=R2
C085      4 NRT(R,LVL)=1
C086      EST=EST1
C087      AEST=AEST1
C088      FB=FM
C089      FT=F3
C090      FM=FS
C091      FS=F1
C092      B=(A+B)/2.000
C093      EPS=EPS/2.000
C094      7 LVL=LVL+1
C095      GO TO 1
C096      11 NRT(R,LVL)=2
C097      FA=FB
C098      FS=F2(LVL)
C099      FM=FTP(LVL)
C100      FT=F4(LVL)
C101      FB=FBP(LVL)
C102      B=XB(LVL)
C103      EST=EST2(LVL)
C104      AEST=AEST2(LVL)
C105      GO TO 7
C106      12 EPS=2.000*EPS
C107      IF(LVL-1)5,5,9
C108      5 ANC4=TSUM-ESUM
C109      RETURN
C110      END

```

4. Adaptive Quadrature Routine Based on Newton-Cotes 7-point Rule

DATE = 68270 12/16/41

FORTRAN IV G LEVEL 1, MCD 1 MAIN

```

C      ANC6 INTEGRATION
C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 7 POINT RULE
C
C001      FUNCTION ANC6(A1,B1,EP,M,A,FUN)
C002      DOUBLE PRECISION ANC6,A1,B1,EP,FUN,A,B,EPS,SUM,ESUM,TSUM,DA,XB,SX
C003      1,F1,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST
C004      21,AEST2,ABSR,DELTA,DIFF,CAFT,FA,SA
C005      DOUBLE PRECISION FX,F5,FU,F6,XR,X5,XU,X6,FUP
C006      DIMENSION F2(30),F4(30),FTP(30),FBP(30),FTST(7),EST2(30),NRTR(30)
C007      DIMENSION AEST2(30),XB(30)
C008      DIMENSION FUP(30),F6(30)
C
C009      C   THE PARAMETER SETUP FOR THE INITIAL CALL
C010      IF(N.LE.0)GC TC 210
C011      IF(N.GT.3)GC TC 211
C012      A=A1
C013      B=B1
C014      EPS=EP*255,CDC
C015      ESUM=C,000
C016      TSUM=C,000
C017      LVL=1
C018      DA=B-A
C019      FA=FUN(A)
C020      FR=FUN((5,000*A+B)/6,000)
C021      FS=FUN((2,000*A+B)/3,000)
C022      FM=FUN((A+B)*C,500)
C023      FT=FUN((A+2,000*B)/3,000)
C024      FU=FUN((A+5,000*B)/6,000)
C025      FB=FUN(B)
C026      M=7
C027      FMAX=DABS(FA)
C028      FTST(1)=FMAX
C029      FTST(2)=DABS(FR)
C030      FTST(3)=DABS(FS)
C031      FTST(4)=DABS(FM)
C032      FTST(5)=DABS(FT)
C033      FTST(6)=DABS(FU)
C034      FTST(7)=DABS(FB)
C035      DO 100 I=2,7
C036      IF(FMAX.GE.FTST(I))GO TO 100
C037      FMAX=FTST(I)
C038      100 CONTINUE
C039      EST=(41,000*(FA+FB)+216,000*(FR+FU)+27,000*(FS+FT)+272,000*FM)*DA/
C040      1840,000
C041      ABSR=(41,000*(FTST(1)+FTST(7))+216,000*(FTST(2)+FTST(6))+27,000*
C042      1FTST(3)+FTST(5))+272,000*FTST(4))*DA/840,000
C043      AEST=ABSR
C044      1=RECUR
C045      1 SX=(DA/(6,000*2,000*LVL))/140,000
C046      1 F1=FUN((11,000*A+B)/12,000)
C047      1 F3=FUN((9,000*A+3,000*B)/12,000)
C048      1 F5=FUN((7,000*A+5,000*B)/12,000)
C049      1 F2(LVL)=FUN((5,000*A+7,000*B)/12,000)
C050      1 F4(LVL)=FUN((3,000*A+9,000*B)/12,000)
C051      1 F6(LVL)=FUN((A+11,000*B)/12,000)
C052      1 EST1=SX*(41,000*(FA+FM)+216,000*(F1+F5)+27,000*(FR+FS)+272,000*F3)
C053      1 FBP(LVL)=FB
C054      1 XB(LVL)=B
C055      1 FTP(LVL)=FT
C056      1 FUP(LVL)=FU
C057      1 EST2(LVL)=SX*(41,000*(FM+FB)+216,000*(F2(LVL)+F6(LVL))+27,000*(FT+
C058      1 F6(LVL))+272,000*F4(LVL))
C059      1 SUM=EST1+EST2(LVL)
C060      1 FTST(1)=DABS(F1)
C061      1 FTST(2)=DABS(F3)
C062      1 FTST(3)=DABS(F5)
C063      1 FTST(4)=DABS(F2(LVL))

```

FORTRAN IV G LEVEL 1, MCD 1

ANC6

DATE = 68270

12/16/41

```

0057      FTST(5)=DABS(F4(LVL))
0058      FTST(6)=DABS(F6(LVL))
0059      FTST(7)=DABS(FM)
0060      AEST1=SX*(41.000*(DABS(FA)+FTST(7))+216.000*(FTST(1)+FTST(3))+27.0
0061      1D0*(DABS(FR)+DABS(FS))+272.000*FTST(2))
0062      AEST2(LVL)=SX*(41.000*(FTST(7)+DABS(FB))+216.000*(FTST(4)+FTST(6))
0063      1+27.000*(DABS(FT)+DABS(FU))+272.000*FTST(5))
0064      ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
0065      M=M+6
0066      GO TO (201,200,202),N
0067      200 DELTA=ABSAR
0068      GO TO 205
0069      210 PRINT 39
0070      39 FORMAT(' ERROR RETURN-N.LE.0')
0071      RETURN
0072      211 PRINT 40
0073      40 FORMAT(' ERROR RETURN-N.GT.3')
0074      RETURN
0075      201 DELTA=1.000
0076      GO TO 205
0077      202 DO 203 I=1,6
0078      IF(FMAX.GE.FTST(I))GO TO 203
0079      FMAX=FTST(I)
0080      203 CONTINUE
0081      DELTA=FMAX
0082      205 DAFT=EST-SUM
0083      DIFT=DABS(DAFT)
0084      DAFT=DAFT/255.000
0085      IF(DIFT-EPS*DELTA)6,6,3
0086      3 IF(LVL-30)4,2,2
0087      6 IF(LVL-1)2,4,2
0088      2=UP
0089      2 A=B
0090      ESUM=FSUM+DAFT
0091      TSUM=TSUM+SUM
0092      9 LVL=LVL-1
0093      L=NRTR(LVL)
0094      GO TC (11,12),L
0095      11=R1,12=R2
0096      4 NRTR(LVL)=1
0097      EST=EST1
0098      AEST=AEST1
0099      FB=FM
0100      FU=FS
0101      FT=F3
0102      FM=F2
0103      FS=FR
0104      FR=F1
0105      B=(A+R1)/2.000
0106      EPS=EPS/2.000
0107      7 LVL=LVL+1
0108      GO TC 1
0109      11 NRTR(LVL)=2
0110      FA=FB
0111      FR=F2(LVL)
0112      FS=F3(LVL)
0113      FM=F4(LVL)
0114      FT=F5(LVL)
0115      FU=F6(LVL)
0116      FB=F7(LVL)
0117      B=XB(LVL)
0118      EST=EST2(LVL)
0119      AEST=AEST2(LVL)
0120      GO TO 7
0121      12 EPS=EPS*2.000
0122      IF(LVL-1)5,9
0123      5 ANC6=TSUM-ESUM
0124      RETURN
0125      END

```

5. Adaptive Quadrature Routine Based on Newton-Cotes 9-point Rule

FORTRAN IV G LEVEL 1, MOD 1	MAIN	DATE = 68270 21/22/06
-----------------------------	------	----------------------------

```

C      ANC8 INTEGRATION
C      ADAPTIVE QUADRATURE RUTINE BASED ON NEWTON-COTES 9 POINT RULE
C
C001    FUNCTION ANC8(A1,B1,EP,M,N,FUN)
C002    DOUBLE PRECISION ANC8,A1,B1,EP,FUN,A,B,EPS,ESUM,TSUM,DA,XB,SX,FA
C003    1,FI,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST
C004    21,AEST2+ABSR,DELTA,DIFF,DAFT,SUM,SA
C005    DOUBLE PRECISION FR,F5,FU,F6,XR,X5,XU,X6,FUP
C006    DOUBLE PRECISION FQ,F7,FV,F8,XQ,X7,XV,X8,FVP
C007    DIMENSION F(20),F4(30),FTP(30),FBP(30),EST2(30),NRTR(30),FTST(9)
C008    DIMENSION AEST2(30),XB(30)
C009    DIMENSION FUP(30),F6(30)
C010    DIMENSION FB(30),FVP(30)
C
C      THE PARAMETER SETUP FOR THE INITIAL CALL
C011    IF(N.LE.0)GO TO 210
C012    IF(N.GT.3)GO TO 211
C013    A=A1
C014    B=B1
C015    EPS=EP*1023.0D0
C016    ESUM=0.0D0
C017    TSUM=0.0D0
C018    LVL=1
C019    DA=B-A
C020    FA=FUN(A)
C021    FQ=FUN((7.0D0*A+B)/8.0D0)
C022    FR=FUN((3.0D0*A+B)/4.0D0)
C023    FS=FUN((5.0D0*A+3.0D0*B)/8.0D0)
C024    FM=FUN((A+B)*5.0D0)
C025    FT=FUN((3.0D0*A+5.0D0*B)/8.0D0)
C026    FU=FUN((A+3.0D0*B)/4.0D0)
C027    FV=FUN((A+7.0D0*B)/8.0D0)
C028    FB=FUN(B)
C029    M=9
C030    FMAX=DABS(FA)
C031    FTST(1)=FMAX
C032    FTST(2)=DABS(FQ)
C033    FTST(3)=DABS(FR)
C034    FTST(4)=DABS(FS)
C035    FTST(5)=DABS(FM)
C036    FTST(6)=DABS(FT)
C037    FTST(7)=DABS(FU)
C038    FTST(8)=DABS(FV)
C039    FTST(9)=DABS(FB)
C040    DO 100 I=2,9
C041    IF(FMAX.GE.FTST(I))GO TO 100
C042    FMAX=FTST(I)
C043    EST=(989.0D0*(FA+FB)+5888.0D0*(FQ+FV)-928.0D0*(FR+FU)+10496.0D0*(F
C044    XS+FT)-4540.0D0*FM)*DA/28350.0D0
C045    ABSAR=(989.0D0*(FTST(1)+FTST(9))+5888.0D0*(FTST(2)+FTST(8))-928.0D0
C046    X0*(FTST(3)+FTST(7))+10496.0D0*(FTST(4)+FTST(6))-4540.0D0*FTST(5))*X
C047    DA/28350.0D0
C
C      AEST=ABSR
C      1=RECUR
C048    1 SX=DA/(28350.0D0*2.0D0**LVL)

```

FORTRAN IV G LEVEL 1, MOD 1

ANC8

DATE = 68270

21/22/06

```

0046      F1=FUN((15.000*A+B)/16.000)
0047      F3=FUN((13.000*A+3.000*B)/16.000)
0048      F5=FUN((11.000*A+5.000*B)/16.000)
0049      F7=FUN((9.000*A+7.000*B)/16.000)
0050      F2(LVL)=FUN((7.000*A+9.000*B)/16.000)
0051      F4(LVL)=FUN((5.000*A+11.000*B)/16.000)
0052      F6(LVL)=FUN((3.000*A+13.000*B)/16.000)
0053      F8(LVL)=FUN((A+15.000*B)/16.000)
0054      EST1=SX*(989.000*(FA+FM)+5888.000*(F1+F7)-928.000*(FQ+FS)+10496.00
X0*(F3+F5)-4540.000*FR)
0055      FBP(LVL)=FB
0056      XB(LVL)=B
0057      FTP(LVL)=FT
0058      FUP(LVL)=FU
0059      FVP(LVL)=FV
0060      EST2(LVL)=SX*(989.000*(FM+FB)+5888.000*(F2(LVL)+F8(LVL))-928.000*(F
XFT+FV)+10496.000*(F4(LVL)+F6(LVL))-4540.000*FU)
0061      SUM=EST1+EST2(LVL)
0062      FTST(1)=DABS(F1)
0063      FTST(2)=DABS(F3)
0064      FTST(3)=DABS(F5)
0065      FTST(4)=DABS(F7)
0066      FTST(5)=DABS(F2(LVL))
0067      FTST(6)=DABS(F4(LVL))
0068      FTST(7)=DABS(F6(LVL))
0069      FTST(8)=DABS(F8(LVL))
0070      FTST(9)=DABS(FM)
0071      AEST1=SX*(989.000*(DABS(FA)+FTST(9))+5888.000*(FTST(1)+FTST(4))-92
X8.000*(DABS(FQ)+DABS(FS))+10496.000*(FTST(2)+FTST(3))-4540.000*DAB
XS(FR))
0072      AEST2(LVL)=SX*(989.000*(FTST(9)+DABS(FB))+5888.000*(FTST(5)+FTST(8))
X)-928.000*(DABS(FT)+DABS(FV))+10496.000*(FTST(6)+FTST(7))-4540.00
X0*DABS(FU))
0073      ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
0074      M=M+8
0075      GO TO (201,20C,202),N
0076      200 DELTA=ABSAR
0077      GO TO 205
0078      210 PRINT 39
0079      39 FORMAT(' ERROR RETURN-N.LE.0')
0080      RETURN
0081      211 PRINT 40
0082      40 FORMAT(' ERROR RETURN-N.GT.3')
0083      RETURN
0084      201 DELTA=1.000
0085      GO TO 205
0086      202 DO 203 I=1,8
0087      IF(FMAX.GE.FTST(I))GO TO 203
0088      FMAX=FTST(I)
0089      203 CONTINUE
0090      DELTA=FMAX
0091      205 DAFT=EST-SUM
0092      DIFF=DABS(DAFT)
0093      DAFT=DAFT/1023.000
0094      IF(DIFF-EPS*DELTA)6,6,3

```

FORTRAN IV G LEVEL 1, MOD 1

ANC8

DATE = 68270

21/22/06

```

0095      3 IF(LVL=30)4,2,2
0096      6 IF(LVL=1)2,4,2
0097      C 2=UP
0098      2 A=B
0099      E SUM=ESUM+DAFT
0100      T SUM=TSUM+SUM
0101      9 LVL=LVL-1
0102      L=NRTR(LVL)
0103      GO TO (11,12),L
0104      C 11=R1,12=R2
0105      4 NRTR(LVL)=1
0106      EST=EST1
0107      AEST=AEST1
0108      FB=FM
0109      FV=F7
0110      FU=FS
0111      FT=FS
0112      FM=FR
0113      FS=F3
0114      FR=FQ
0115      FQ=F1
0116      7 B=(A+B)/2.000
0117      EPS=EPS/2.000
0118      11 LVL=LVL+1
0119      GO TO 1
0120      11 NRTR(LVL)=2
0121      FA=FB
0122      FQ=F2(LVL)
0123      FR=FTP(LVL)
0124      FS=F4(LVL)
0125      FM=FUP(LVL)
0126      FT=F6(LVL)
0127      FU=FVP(LVL)
0128      FV=F8(LVL)
0129      FB=FBP(LVL)
0130      B=XB(LVL)
0131      EST=EST2(LVL)
0132      AEST=AEST2(LVL)
0133      GO TO 7
0134      12 EPS=2.000*EPS
0135      IF(LVL=1)5,5,9
0136      5 ANC8=TSUM-ESUM
0137      RETURN
0138      END

```

6. Adaptive Quadrature Routine Based on Newton-Cotes 11-point Rule

```

FORTRAN IV G LEVEL 1, MOD 1           MAIN          DATE = 68270      21/22/06

C      ANC10 INTEGRATION
C      ADAPTIVE QUADRATURE ROUTINE BASED ON NEWTON-COTES 11 POINT RULE
C
0001   FUNCTION ANC10(A1,B1,FP,M,N,FUN)
0002   DOUBLE PRECISION ANC10,A1,B1,EP,FUN,A,B,EPS,SUM,ESUM,TSUM,DA,XB,S
0003   1X,F1,F5,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AES
0004   2T1,AEST2,ABSAR,DELTA,CIFF,DAFT,FA,SA
0005   DOUBLE PRECISION FR,F5,FU,F6,XR,X5,XU,X6,FUP
0006   DOUBLE PRECISION FQ,F7,FV,F8,XQ,X7,XV,X8,FVP
0007   DOUBLE PRECISION FP,F9,FW,F10,XP,X9,XW,X10,FWP
0008   DIMENSION F2(30),F4(30),FTP(30),FBP(30),EST2(30),NRTR(30),FTST(11)
0009   DIMENSION AEST2(30),XB(30)
0010   DIMENSION FUP(30),F6(30)
0011   DIMENSION F8(30),FVP(30)
0012   DIMENSION F10(30),FWP(30)
C     THE PARAMETER SETUP FOR THE INITIAL CALL
0013   IF(N.LE.0)GO TO 210
0014   IF(N.GT.3)GO TO 211
0015   A=A1
0016   B=B1
0017   EPS=EP*4095.000
0018   ESUM=0.000
0019   TSUM=0.000
0020   LVL=1
0021   DA=B-A
0022   FA=FUN(A)
0023   FP=FUN((9.000*A+B)/10.000)
0024   FQ=FUN((4.000*A+B)/5.000)
0025   FR=FUN((7.000*A+3.000*B)/10.000)
0026   FS=FUN((3.000*A+2.000*B)/5.000)
0027   FM=FUN((A+B)*0.500)
0028   FT=FUN((2.000*A+3.000*B)/5.000)
0029   FU=FUN((3.000*A+7.000*B)/10.000)
0030   FV=FUN((A+4.000*B)/5.000)
0031   FW=FUN((A+9.000*B)/10.000)
0032   FB=FUN(B)
0033   M=11
0034   FMAX=DABS(FA)
0035   FTST(1)=FMAX
0036   FTST(2)=DABS(FP)
0037   FTST(3)=DABS(FQ)
0038   FTST(4)=DABS(FR)
0039   FTST(5)=DABS(FS)
0040   FTST(6)=DABS(FM)
0041   FTST(7)=DABS(FT)
0042   FTST(8)=DABS(FU)
0043   FTST(9)=DABS(FV)
0044   FTST(10)=DABS(FW)
0045   FTST(11)=DABS(FB)
0046   DO 100I=2,11
0047   IF(FMAX.GE.FTST(I))GO TO 100
0048   100 CONTINUE
0049   EST=(16067.000*(FA+FB)+106300.000*(FP+FW)-48525.000*(FQ+FV)+272400
0050   X.000*(FR+FU)-260550.000*(FS+FT)+427368.000*FM)*DA/598752.000

```

```

FORTRAN IV G LEVEL 1, MOD 1           ANC10          DATE = 68270        21/22/68
0049          ABSAR=(16067.000*(FTST(1)+FTST(11))+106300.000*(FTST(2)+FTST(10))-X48525.000*(FTST(3)+FTST(9))+272400.000*(FTST(4)+FTST(8))-260550.00X0*(FTST(5)+FTST(7))+427368.000*FM)*DA/598752.000
0050          AEST=ABSAR
0051          C      1=RECUR
0052          1  SX=DA/(598752.000*2.000**LVL)
0053          F1=FUN((19.000*A+B)/20.000)
0054          F3=FUN((17.000*A+3.000*B)/20.000)
0055          F5=FUN((15.000*A+5.000*B)/20.000)
0056          F7=FUN((13.000*A+7.000*B)/20.000)
0057          F9=FUN((11.000*A+9.000*B)/20.000)
0058          F2(LVL)=FUN((9.000*A+11.000*B)/20.000)
0059          F4(LVL)=FUN((7.000*A+13.000*B)/20.000)
0060          F6(LVL)=FUN((5.000*A+15.000*B)/20.000)
0061          F8(LVL)=FUN((3.000*A+17.000*B)/20.000)
0062          F10(LVL)=FUN((A+19.000*B)/20.000)
0063          EST1=SX*(16067.000*(FA+FM)+106300.000*(F1+F9)-48525.000*(FP+FS)+27X2400.000*(F3+F7)-260550.000*(FQ+FR)+427368.000*F5)
0064          FBP(LVL)=FB
0065          XB(LVL)=B
0066          FTP(LVL)=FT
0067          FUP(LVL)=FU
0068          FVP(LVL)=FV
0069          FWp(LVL)=FW
0070          EST1=EST1+EST2(LVL)
0071          FTST(1)=DABS(F1)
0072          FTST(2)=DABS(F3)
0073          FTST(3)=DABS(F5)
0074          FTST(4)=DABS(F7)
0075          FTST(5)=DABS(F9)
0076          FTST(6)=DABS(F2(LVL))
0077          FTST(7)=DABS(F4(LVL))
0078          FTST(8)=DABS(F6(LVL))
0079          FTST(9)=DABS(F8(LVL))
0080          FTST(10)=DABS(F10(LVL))
0081          FTST(11)=DABS(FM)
0082          AEST1=SX*(16067.000*(DABS(FA)+FTST(11))+106300.000*(FTST(1)+FTST(5X))-48525.000*(DABS(FP)+DABS(FS))+272400.000*(FTST(2)+FTST(4))-2605X0.000*(DABS(FQ)+DABS(FR))+427368.000*FTST(3))
0083          AEST2(LVL)=SX*(16067.000*(FTST(11)+DABS(FB))+106300.000*(FTST(6)+FTST(10))-48525.000*(DABS(FT)+DABS(FW))+272400.000*(FTST(7)+FTST(9))-260550.000*(DABS(FU)+DABS(FV))+427368.000*FTST(8))
0084          ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)
0085          M=M+10
0086          GO TO (201,200,202),N
0087          200 DELTA=ABSAR
0088          GO TO 205
0089          210 PRINT 39
0090          39 FORMAT(' ERROR RETURN-N.LE.0')
0091          RETURN
0092          211 PRINT 40
0093          40 FORMAT(' ERROR RETURN-N.GT.3')

```

FORTRAN IV G LEVEL 1, MCD 1

ANC10

DATE = 68270

21/22/06

```

0094      RETURN
0095 201 DELTA=1.000
0096      GO TO 205
0097 202 DO 203 I=1,10
0098      IF(FMAX.GE.FTST(I))GO TO 203
0099      FMAX=FTST(I)
0100 203 CONTINUE
0101      DELTA=FMAX
0102 205 DAFT=EST-SUM
0103      DIFF=DABS(DAFT)
0104      DAFT=DAFT/4095.000
0105      IF(DIFF-EPS*DELTAI6,6,3
0106      3 IF(LVL-3)4,2
0107      6 IF(LVL-1)2,4,2
0108      2=UP
0109      2 A=B
0110      ESUM=ESUM+DAFT
0111      TSUM=TSUM+SUM
0112      9 LVL=LVL-1
0113      L=NRTR(LVL)
0114      GO TO (11,12),L
0115      11=R1,12=R2
0116      4 NRTR(LVL)=1
0117      EST=EST1
0118      AEST=AEST1
0119      FB=FM
0120      FW=F9
0121      FS=FS
0122      FU=F7
0123      FT=FR
0124      FM=F5
0125      FS=FQ
0126      FR=F3
0127      FQ=FP
0128      FP=F1
0129      B=(A+B)/2.000
0130      EPS=EPS*2.000
0131      7 LVL=LVL+1
0132      GO TO 1
0133      11 NRTR(LVL)=2
0134      FA=FB
0135      FP=F2(LVL)
0136      FQ=FTP(LVL)
0137      FR=F4(LVL)
0138      FS=FUP(LVL)
0139      FM=F6(LVL)
0140      FT=FVP(LVL)
0141      FU=F8(LVL)
0142      FW=F9P(LVL)
0143      FB=FBP(LVL)
0144      B=XB(LVL)
0145      EST=EST2(LVL)
0146      AEST=AEST2(LVL)
0147      GO TO 7
0148      12 EPS=EPS*2.000
0149      IF(LVL-1)5,5,9
0150      5 ANC10=TSUM-ESUM
0151      RETURN
END

```

REFERENCES

1. McKeeman, W. M., Algorithm 145, *Adaptive Numerical Integration by Simpson's Rule*, Comm. ACM, Vol. 5 (1962), p. 604.
2. McKeeman, W. M., and Tesler, L., Algorithm 182, *Nonrecursive Adaptive Integration*, Comm. ACM, Vol. 6 (1963), p. 315.
3. McKeeman, W. M., Algorithm 198, *Adaptive Integration and Multiple Integration*, Vol. 6 (1963), p. 443.
4. Davis, P., and Rabinowitz, P., *Numerical Integration*, Blaisdell Publ. Co., London, 1967.
5. Lyness, J. N., *Notes on the Adaptive Simpson Quadrature Routine*, submitted to JACM.

ARGONNE NATIONAL LAB WEST



3 4444 00008205 7

X